**Unity Certification Preparation:**

**C# Programming**

**Orlando Unity3d Development Meetup**

**Date: 24-Sep-18**

**All Copyright PompaSetting Studios © 2018**

**Contents**

[**6.0** **Destroying Objects** 3](#_Toc524360280)

[**6.1** **Create A New Project** 3](#_Toc524360281)

# **6.0 Destroying Objects**

One of the functions we are going to use is called Destroy(); > Destroy() does the opposite to creating or instantiating an object. Destroy removes the object from the scene (i.e. destroys the gameObject to which the script is attached). So, in the start function we write Destroy Open parentheses ( and pass the parameter…the game object)

Destroy (gameObject > gameObject = the object to which this script is attached, then if we type comma, then one can pass multiple parameters:

* **1st parameter = the gameObject,**
* **2nd parameter = the TIME we want it to be destroyed**

Therefore, if we want the cube to be destroyed after 3 seconds, we just write 3f. (i.e. Destroy(gameObject, 3f); > then add a closing parenthesis, and a semi-colon; …too recap after the game starts the object to which the script is attached is destroyed after 3 seconds approximately. For the sake of completeness, we add a debug.log ("Cube is destroyed !"); (to output a message log) > Save then exit the script. Ok so now if we go back to our editor in unity, lets create a 3d object cube game object as follows. So now if we run this in game mode. What one should see in the game view is the cube appear then disappears in 3 seconds.

# **6.1 Create A New Project**

|  |
| --- |
| **CREATE A 1 x NEW PROJECT > 2 x OBJECTS, 2 x MATERIALS, 1 x SCRIPT** |
| [Navigate to](Click on)[UNITY ICON] > [Navigate to](Click)[New] > [Navigate to ][Project Name]{Project Name Placeholder} < TESTX001 > |
| [Navigate to][Location]{Location Placeholder}'(Record the location) < 'C:\Users\SeanA\Desktop\\_UNITYPROJS\ > |
| [Navigate to][Create Project](Click) |
|  |
| **CREATE 3 NEW ASSET FOLDERS: 1 x\_MATERIALS, 1 x\_SCRIPTS, 1 x\_SCENES, 1 x \_PREFABS** |
| (Navigate to PROJECT tab)[ASSETS](RIGHT CLICK)(Select)[CREATE][Folder][F2](Rename) < \_MATERIALS > |
| (Navigate to PROJECT tab)[ASSETS](RIGHT CLICK)(Select)[CREATE][Folder][F2](Rename) < \_SCRIPTS > |
| (Navigate to PROJECT tab)[ASSETS](RIGHT CLICK)(Select)[CREATE][Folder][F2](Rename) < \_SCENES > |
| [Navigate to PROJECT tab][ASSETS](RIGHT CLICK)(Select)[CREATE][Folder][F2](Rename) < \_PREFABS > |
|  |
| **CREATE 1 OBJECT > 1 x PLANE** |
| (Navigate to)(Select)[SCENE window] |
| (Navigate) [Hierarchy tab] > (RIGHT CLICK)(Select) [3D OBJECT] [PLANE] |
| > (Navigate to)[INSPECTOR tab] > [Navigate to][TRANSFORM][SCALE][X] < 5 >, [Y] < 5 >, [Z] < 5 > |
|  |
| **CREATE 2 OBJECT > 1 x CAPSULE** |
| (Navigate) [Hierarchy tab] > (RIGHT CLICK)(Select) [3D OBJECT] [PLANE][CAPSULE] |
| > (Navigate to)[INSPECTOR tab] > [Navigate to][TRANSFORM][POSITION][X] < 1 >, [Y] < 1.5 >, [Z] < 1 > |
| > (Navigate to)[INSPECTOR tab] > [Navigate to][TRANSFORM][ROTATION] > [X] = [Y] = [Z] < 0 > |
| > (Navigate to)[INSPECTOR tab] > [Navigate to][TRANSFORM][SCALE][X] < 1 >, [Y] < 1 >, [Z] < 1 > |
|  |
| **CREATE 2 x MATERIALS** |
| (Navigate to)[PROJECT tab]> (Select)[ASSETS][MATERIALS] > (RIGHT CLICK)[CREATE][MATERIALS] > (F2 Rename) < RED >, < R = 255, G = B = 0, A = 255 > |
| (Navigate to)[PROJECT tab]> (Select)[ASSETS][MATERIALS] > (RIGHT CLICK)[CREATE][MATERIALS] > (F2 Rename) < GREEN >, < R = 0, G = 255, B = 0, A = 255 > |
| (Navigate to)[PROJECT tab]> (Select)[ASSETS][MATERIALS] > (Select)(Drag)[RED] > (Navigate to)[HIERARCHY tab](DROP)[CAPSULE] |
| (Navigate to)[PROJECT tab]> (Select)[ASSETS][MATERIALS] > (Select)(Drag)[GREEN] > (Navigate to)[HIERARCHY tab](DROP)[PLANE] |
|  |
| **CREATE A NEW 1 x SCRIPT** |
| Navigate to PROJECT tab > Select ASSETS folder > Select SCRIPTS folder > [RIGHT CLICK] Create > C# Script > F2 RENAME or NAME = "NewScriptX" |
| Now add our test script to Cube1 suspended in mid-air. Add Components > Scripts > CollisionScript |
| Navigate to PROJECT tab > Drag N Drop NewScriptX onto > CUBE gameObject (inside the HIERARCHY tab) |

|  |
| --- |
| using System.Collections; |
| using System.Collections.Generic; |
| using UnityEngine; |
|  |
| public class NewScript : MonoBehaviour |
| { |
| // Use this for initialization |
| **void Start ()** |
| **{** |
| **Destroy(gameObject, 3f);** |
| **Debug.Log("The CAPSULE is now destroyed !");** |
| **}** |
|  |
| // Update is called once per frame |
| void Update () |
| { |
| } |
| }//End of NewScript |
|  |

We can call this destroy function from wherever we want. Later on, we will learn how to destroyed different gameObjects.

/End